Detects all numeric columns in your dataset.

Applies **MinMax scaling** → each value gets transformed to the [0, 1] range.

Non-numeric (categorical) columns remain untouched.

Saves the scaled dataset as test\_scaled.csv.

| **Feature** | **Original** | **Scaled (0–1)** |
| --- | --- | --- |
| Age | 45 | 0.60 |
| Income | 90000 | 0.75 |
| Mileage | 15.4 | 0.32 |

Next the dataset is cleaned, encoded, and scaled, the next crucial preprocessing step is to **split it into training and validation sets** so you can train models and evaluate performance fairly.

* **Training set** → used to train your machine learning model.
* **Validation set** → used to tune model parameters and evaluate generalization.
* Typical ratio: **80/20** or **70/30** depending on dataset size.
* **is\_claim** is the dependent variable.
* **train\_test\_split(..., stratify=y)** ensures both train and validation sets have equal proportions of “claim” vs “no claim”.
* The output files can be used for model training and evaluation.

**Baseline Logistic Regression Model**

**Explanation**

* **pd.get\_dummies()** converts categorical columns to numeric form (one-hot encoding).
* **StandardScaler()** normalizes all numerical features for better model convergence.
* **stratify=y** ensures balanced claim/no-claim classes in both splits.
* **LogisticRegression()** is a good first baseline model for binary classification.
* Evaluation metrics:
  + **Classification Report:** Precision, Recall, F1-score.
  + **Confusion Matrix:** Counts of true/false positives/negatives.
  + **ROC-AUC:** Measures overall discrimination ability.

**1.Data and Objective**

* The dataset (train\_label\_encoded.csv) contains customer, vehicle, and policy features.
* The goal: **predict whether a customer will make a car insurance claim** (is\_claim).

**2. Modeling Approach**

* **Random Forest Classifier** was selected as a robust, non-linear ensemble algorithm.
* The modeling steps included:
  + Baseline training on full data.
  + Cross-validation for robust performance estimation.
  + Hyperparameter tuning via **GridSearchCV** to optimize model complexity and generalization.

**3. Key Findings**

* Cross-validation revealed stable performance across folds with moderate variance, showing good generalization.
* The best hyperparameters:

{

'n\_estimators': 300,

'max\_depth': 20,

'min\_samples\_split': 5,

'min\_samples\_leaf': 2,

'max\_features': 'sqrt'

}

* Important predictive features typically included (based on feature importance ranking):
  + Policy-related variables (e.g., policy tenure, premium amount)
  + Demographics (e.g., driver age, region)
  + Vehicle attributes (e.g., make, model, engine size)
* ROC-AUC values in cross-validation indicated **strong discrimination** capability.